Lab Assignment

Cybersecurity Professional Program

Introduction to Python for Security

Data Types & Conditions

PY-02-L5
IP Address to Binary Converter

Lab Objective

Understand how to perform string operations and data type conversions, and get introduced to basic list operations.

Lab Mission

Practice working with inputs and string operations to generate different outputs.

Lab Duration

15-20 minutes.

Requirements

- Knowledge of how to handle inputs from the user.
- Working knowledge of variables.
- Working knowledge of string and collection manipulation.

Resources

- Environment & Tools
    - Windows
        - Python 3
        - PyCharm

Textbook References

- Chapter 2: Data Types & Conditions

- Section 1: Variables and User Output
- Section 5: String Manipulation

Lab Task

Construct an interactive script that will ask the user for an IP address, and convert it to a binary value.

The output will be the octet in decimal format, and the octet in binary.

Example: 192: 11000000

Note: In this lab you will get familiar with a new function **bin()** that converts an integer to its binary representation.

```
print(bin(10))

0b1010
# 0b represent base 2 value, 1010 is the binary result.
```

1 Create a variable to ask the user for an IP address.

```
ip_address = input("Please enter an IP address: ")
```

2 Split each octet using "." as a delimiter, and store it in a new variable.

```
ip_octets = ip_address.split(".")
```

3 Print the result of the splitting.

```
print(ip_octets)
```

4 Store the first octet in a variable as an integer.

```
First = int(ip_octets[0])
```

5 Store the second octet in a variable as an integer.

```
Second = int(ip_octets[1])
```

6 Store the third octet in a variable as an integer.

```
Third = int(ip_octets[2])
```

7 Store the fourth octet in a variable as an integer.
Note: The function **bin()** converts a number to its binary representation.

```
Fourth = int(ip_octets[3])
```

8 Print the first octet's binary value.
Note: The function **bin()** converts a number to its binary representation.

```
print("{} : {}".format(First, bin(First)[2:]))
```

9 Print the second octet's binary value.
Note: The function **bin()** converts a number to its binary representation.

```
print("{} : {}".format(Second, bin(Second)[2:]))
```

10 Print the third octet's binary value.
Note: The function **bin()** converts a number to its binary representation.

```
print("{} : {}".format(Third, bin(Third)[2:]))
```

11 Print the fourth octet's binary value.
Note: The function **bin()** converts a number to its binary representation.

```
print("{} : {}".format(Fourth, bin(Fourth)[2:]))
```